IN THE SPECIFICATION:

Please replace the first full paragraph on page 3 in the specification with the following replacement paragraph:

— Disk storage is typically implemented as one or more storage "volumes" that comprise physical storage disks, defining an overall logical arrangement of storage space. Currently available filer implementations can serve a large number of discrete volumes (150 or more, for example). Each volume is associated with its own file system and, for purposes hereof, volume and file system shall generally be used synonymously. The disks within a volume are typically organized as one or more groups of Redundant Array of Independent (or *Inexpensive*) Disks (RAID). RAID implementations enhance the reliability/integrity of data storage through the redundant writing of data "stripes" across a given number of physical disks in the RAID group, and the appropriate caching of parity information with respect to the striped data. In the example of a WAFL file system, a RAID 4 implementation is advantageously employed. This implementation specifically entails the striping of data across a group of disks, and separate parity caching within a selected disk of the RAID group. As described herein, a *volume* typically comprises at least one data disk and one associated parity disk (or possibly ~~data/parity)~~ data/parity partitions in a single disk) arranged according to a RAID 4, or equivalent high-reliability, implementation. —

Please replace the first full paragraph on page 4 in the specification with the following replacement paragraph:

— If the parity blocks are all stored on one disk, thereby providing a single disk that contains all (and only) parity information, a RAID-4 implementation is provided. If the parity blocks are contained within different disks in each stripe, usually in a rotating

pattern, then the implementation is RAID-5. If one disk fails in the parity group, the contents of that disk can be *reconstructed* on a second "spare" disk or disks by adding all the contents of the remaining data blocks and subtracting the result from the parity block. Since two's ~~compliment~~ complement addition and subtraction over one-bit fields are both equivalent to XOR operations, this reconstruction consists of the XOR of all the surviving data and parity blocks. Similarly, if the parity disk is lost, it can be recomputed in the same way from the surviving data. —